Learning Robotic Tool-Screw Misalignment from Force Feedback

April 29, 2025

Primary Author: Art Boyarov (aboyarov@umich.edu)
Advised by: Yating Lin (yatinlin@umich.edu) and Dr. Dmitry Berenson (dmitryb@umich.edu)
All individuals affiliated with the Department of Robotics at the University of Michigan

***Abstract*—Screw fastening is a key task in robotic assembly and disassembly. In unstructured environments with screws at previously unknown positions, computer vision is used to localize screws, but these pose estimates usually have errors exceeding manufacturing tolerances. This work builds on previous research to use contact force information when fastening the screw to correct the misalignment due to incorrect pose estimates. Previous work falls short due to strict assumptions about the screw pose, the need for time-intensive data collection, and a lack of attention paid to complex geometries, such as a hexagonal screw. In this work, we first develop a generalized accommodation controller capable of limiting contact forces during screw fastening. Second, a simulation of the screw fastening operation is produced and used to collect synthetic training data. Third, a multi-layer perceptron network is trained to predict misalignment from the contact wrench (a stacked set of forces and torques), joint angles, and estimated screw pose during fastening. Lastly, the network is integrated into a controller executing the whole fastening and misalignment correction operation. Success rates of 97.5% and 99% are achieved when correcting misalignment when fastening a screw at a fixed and variable position between trials respectively. However, when both the position and rotation of the screw are varied, the controller fails the misalignment correction task. These results show our method can learn the contact wrench to misalignment relationship for the hexagonal screw's complex geometry. Additionally, the success with an unfixed screw position shows potential for our method in unstructured assembly and disassembly tasks. However, we see than even more data is needed to learn the misalignment relationship when the screw's rotation is varied. Further work is required to find ways to extend our method without simply increasing the amount of training data.**

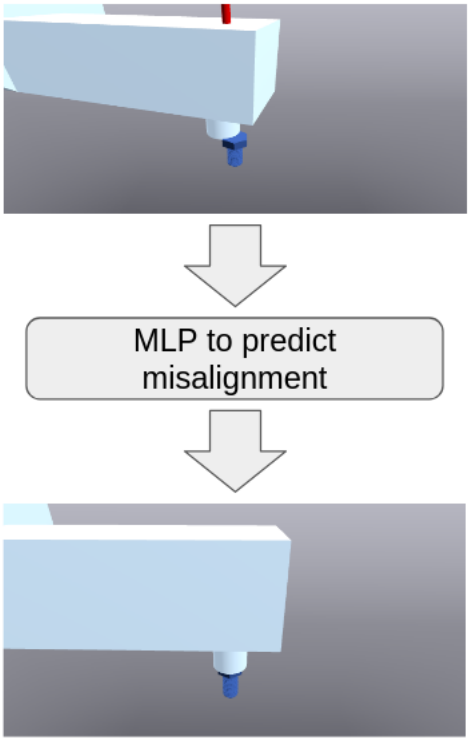

Fig. 1. Our proposed method: using a MLP network to learn the contact wrench to screw misalignment relationship.

I. Introduction

A. Background

Robots are widely used in industrial applications for assembly and disassembly tasks, in which screw fastening is a key operation [1]. Novel applications such as electric vehicle battery repair, e-waste disassembly, and flexible manufacturing require robots to handle fastening tasks in unstructured environments, where screw poses are unknown prior to assembly or disassembly [1]. Screws are localized using vision-based systems, although state-of-the-art pose estimation techniques have errors on the order of several millimeters, which exceeds manufacturing tolerances [2]. Therefore, additional information during the screw fastening process is needed to more accurately localize the screw, or correct the misalignment induced by an incorrect pose estimate. Since the fastening task is an instance of contact-rich manipulation [3], contact forces during fastening have been used as additional information to guide the misalignment correction process [1].

B. Previous Work

In previous work, researchers have approached the broader problem of peg-in-hole fastening, where a cylindrical peg must be inserted into a circular hole. This problem is representative of the screw fastening task - during screw fastening, the screw bit must be aligned with and inserted onto the screw head, similarly to how the peg must be aligned with and inserted into the hole [4].The most basic approach to use contact forces to correct misalignment is using a remote center of compliance: a mechanical device that deforms elastically upon contact in order to align the peg with the hole [5]. This is an example of passive compliance, where the robot's mechanism

allows it to change its motion under external forces. The main disadvantages of this device are its lack of adaptability to applications aside from peg-in-hole tasks, and the risk of mechanical failure in the compliance mechanism [1].

Active compliance strategies involve a control algorithm automatically adjusting the robot's path due to contact forces, as opposed to using a built-in mechanism in the robot. In this case, contact forces are obtained from a force-torque sensor on the robot's end effector, or from torques measured at the robot's joints [6]. In [7], the authors derived an analytical expression to calculate the peg misalignment from the contact force. However, this work's drawbacks are the assumption that the peg has no misalignment along the axis of approach to the hole, which is not always the case. Additionally, this work assumes a quasi-static insertion operation, which does not apply with high speeds, and the effects of friction at high speed make the analytical model invalid [7]. Recently, learning-based approaches have been explored to learn the contact force to misalignment relationship through experimental data. In [8], the authors use collect training data consisting of the contact wrench when inserting a misaligned peg at different, known misalignments. Then, they use supervised learning to learn a relationship between the contact wrench and misalignment. This method is incredibly successful, with a 99% success rate on correcting unknown misalignments in a real-world insertion task. Another successful approach to correcting misalignment for peg-in-hole tasks involves reinforcement learning: in [9], the authors trained a recurrent neural network implemented as an LSTM using reinforcement learning and were able to achieve 100% success rates fastening pegs with a clearance of 6 μm. However, the main issues with these works is they focus on the simpler peg-in-hole task and do not generalize to complex geometry peg-in-hole tasks, where the irregular geometry of the peg introduces a rotational misalignment that must be corrected. Additionally, these works required training data to be collected through a time-consuming process on a physical robot. In [8], collecting 1000 data points took over 8 hours. These drawbacks limit the ability of these methods to be used in industrial settings, where fastening tasks involve shapes of geometries more complex than a cylindrical peg [10], and where adapting to new assembly operations and gathering new training data if necessary must be done quickly with low downtime [1].

Another important aspect of the peg-in-hole and screw fastening operations is the need for compliant controllers for end effector pose tracking. When controlled by a pure position controller, a robot will attempt to forcefully push through the screw when it comes into contact [8]. This will lead to the screw bit and screw head breaking. A compliant controller is capable of limiting forces exerted by the robot during contact, thereby stopping the robot from damaging itself or its environment. A common compliant controller is an impedance controller, which treats the robot as a mass-spring-damper system. When the robot comes into contact, the mass, damping ratio, and stiffness dictate the time taken to come to a stop and the resultant contact force caused by the displacement induced by coming into contact ahead of a desired pose [11]. The impedance controller is a popular choice across many existing works in correcting peg-in-hole misalignment [9, 8].

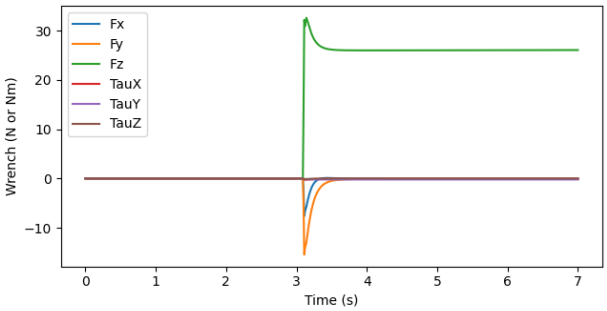

Fig. 2. The contact wrench during the screw fastening operation for a misaligned screw.

C. Project Contributions

This work aims to extend previous research on learning a contact force to misalignment relationship for peg-in-hole fastening by extending the problem to fastening hexagonal screws. Additionally, the possibility of collecting synthetic training data to learn a force to misalignment relationship is explored. In this work, first, a generalized accommodation controller (GAC) is implemented and used to control a robot during the screw fastening task. Second, a simulation of the screw fastening task using the Drake libraries is developed and used to collect data for the contact wrench when fastening a misaligned screw. Third, a multi-layer perceptron (MLP) neural network is trained to predict the screw misalignment from the contact wrench, joint angles, and final end effector pose, similar to the method used in [8]. Lastly, the trained network is integrated into a controller and deployed on a screw fastening task. In each of the tasks, we change how the location of the screw varies: in the first task, the screw is at a fixed position and rotation. In the second task, the screw rotation is fixed, but the 2D position is varied in a 10cm by 10cm envelope. In the third task, the screw rotation and position are varied. In all tasks, the controller is given a noisy estimate of the screw pose and must correct this estimate using the contact wrench during fastening.

II. Methods

A. Generalized Accommodation Control

Due to the contact-rich nature of the screw fastening task, a controller capable of limiting contact forces upon contact with the screw is required. Previous works have used an impedance controller, which treats the robot as a mass-spring-damper system, to ensure safe behavior during contact [4, 9]. However, this impedance controller cannot safely handle abrupt collisions far from where contact is expected, as the contact force is proportional to the displacement during contact from the desired pose. Additionally, impedance control features inherent

trade-offs between precise trajectory following and low contact forces. A higher stiffness gain in the impedance controller improves trajectory tracking, but at the cost of high contact forces [12].

A generalized accommodation controller (GAC), as proposed in [8] and [12], is able of achieving these two objectives: precise trajectory tracking and low contact forces. This work implements a GAC similar to the non-linear accommodation controller presented in [12]. The discrete-time equation for the GAC used in this work is:

$$X_c[k] = X_c[k-1] + \exp(-K_a f_{ext})\Delta X_r \tag{1}$$

X_c	Commanded pose to cartesian impedance controller
ΔX_r	Discrete-time trajectory step
f_{ext}	Contact force
K_a	Gain

The commanded pose $X_c[k]$ is sent to a cartesian impedance controller which converts the cartesian-space pose command to a set of joint torques as follows [13]:

$$\begin{aligned} F_d &= K(x - x_d) + B(\dot{x} - \dot{x}_d) \\ \tau_{ctrl} &= J^T \Lambda F_d + c(q, \dot{q}) + g(q) \end{aligned} \tag{2}$$

K	Stiffness coefficients
B	Damping coefficients
$x, \dot{x}$	Cartesian end-effector pose and velocity
J	Manipulator Jacobian
Λ	Manipulator kinetic energy matrix
$c(q, \dot{q})$	Manipulator coriolis matrix
$g(q)$	Gravity compensation torques

This GAC is able to achieve precise trajectory tracking while limiting contact forces. During a screw fastening operation, the contact wrench converges within 2 seconds and has contact forces all lower than 40 newtons. Figure 2 shows the contact wrench over time during the screw fastening operation.

B. Collecting Training Data

To simulate the screw fastening operation, the Drake robotic modeling and simulation framework was used [14]. Drake is capable of high-fidelity contact modeling, which is key to getting accurate contact forces during the screw fastening operation. Drake uses the hydroelastic contact model, which models contacting objects as compliant (deformable) materials, to solve for reaction forces during contact [15]. In each step of the data collection simulation, offsets in the x,y, and rotational directions were calculated and added to the screw's 2D position and rotation to produce a misaligned screw position replicating an erroneous pose estimate from a computer vision system. The misaligned screw pose was then used to build a fastening trajectory with the robot arm's end effector starting directly above the misaligned screw position estimate and ending at the misaligned position. The robot arm used the aforementioned GAC to follow the fastening trajectory and come in to contact with the screw. The robot end effector took 4 seconds to travel the fastening trajectory. After contact, the simulator waited 2 seconds for the contact wrench, robot joint angles, and end effector wrench to converge. The final contact wrench, joint angles, end effector pose, and misalignment were then recorded as one data point.

C. Supervised Learning of Misalignment

A MLP neural network was trained to learn the screw misalignment from the robot's joint angles, contact wrench, and robot end effector pose during fastening. The joint angles were given as an input to the network. For a compliant controller, the end effector contact wrench depends on the robot joint configuration [13]. Therefore, adding the joint angles enables the network to better learn the misalignment to contact wrench relationship. Additionally, the robot end effector pose was given as an input to the network to improve learning the misalignment. Previous works in correcting misalignment for peg-in-hole tasks also mention inputting the robot end effector pose during fastening to improve learning of the contact wrench to misalignment relationship [9].

The network had 3 layers, with 200 neurons on both hidden layers, and tanh activations to capture nonlinearities in the relationship between contact wrench and misalignment. The MLP structure, training, and testing was implemented in PyTorch. In the MLP, linear layer weights were initialized using Xavier initialization using PyTorch's `torch.nn.init.xavier_uniform_` function. This was done to improve learning and to enable random initialization for networks in an ensemble. While testing the full screw fastening task, ensembles were used to check if data gathered in the full fastening task was out of distribution. The loss function for the estimated misalignment $[\hat{x} \quad \hat{y} \quad \hat{\theta}]$ (the output of the networks) is given in Equation 3. The screw radius is r, and $MSE(a, b)$ is the mean-squared-error loss. Additionally, L2 norm regularization was added as a term to the loss function to reduce overfitting of the network.

$$\begin{aligned} \mathcal{L}(x, y, \theta, \hat{x}, \hat{y}, \hat{\theta}) = MSE(x, \hat{x}) + MSE(y, \hat{y}) \\ + r \cdot MSE(\theta, \hat{\theta}) \end{aligned} \tag{3}$$

The network was trained separately for the three different fastening tasks. For each task, the network was trained on a different dataset specific to each task. For the task with a fixed screw pose, more misalignments were collected in the dataset to better learn the contact wrench to misalignment relationship. For the other tasks, having more trials with screws in different positions meant the number of different misalignments had to be reduced, in order to complete the data collection within a reasonable amount of time, but to enable learning the misalignment relationship for different screw poses and joint angles.

Dataset name	Screw X positions	Screw Y positions	Screw Angular Positions	Misalignments in X direction	Misalignments in Y direction	Angular misalignments	Time to collect data (hrs)
`fixed-xyt-l`	1	1	1	41	41	41	8.5
`fixed-xyt-s`	1	1	1	11	11	11	0.1
`unfixed-xy-xyt-l`	1	9	9	21	21	21	11
`unfixed-xy-xyt-s`	1	4	4	5	5	5	0.5
`unfixed-xyt-xyt-l`	5	4	4	15	15	15	27.0
`unfixed-xyt-xyt-s`	3	3	3	3	3	3	0.1

TABLE I
DATASETS COLLECTED FOR LEARNING A SCREW MISALIGNMENT MODEL.
NAMES FORMED AS `{Screw pose: fixed, unfixed}-{Unfixed screw poses: xy, xyt}-{xyt}-{Large (l) or small (s)}`.

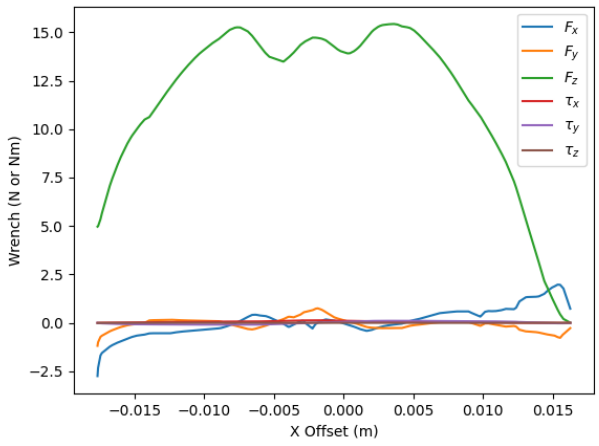

Fig. 3. The contact wrench when fastening with different offsets in the x direction.

III. RESULTS

A. Collecting Training Data

In the simulation, a hexagonal screw with head diameter of 14.3 mm and length of 19.05 mm was used. A KUKA iiwa 7 robot arm with a wrench secured in a gripper was used to model the robot carrying out the fastening operation. The hex bit on the wrench had a diameter of 15.8 mm , thereby leaving a tolerance of roughly 1.5 millimeters between the screw head and bit. The robot arm was oriented with the base extending in the positive x direction, as this is how our lab's robot is secured.

The range of offsets in the x and y directions was between -15 and 15 mm and the range of angular offsets was between -30 and 30. Due to the rotational symmetry of the hexagonal screw, an angular misalignment is the same every 60, meaning a range of 60 for the angular misalignments is enough to cover all possible angular misalignments. For the large datasets, offsets and screw positions were calculated by interpolating linearly between the maximum and minimum bounds. For the small datasets, offsets and screw positions were randomly sampled from the specified ranges.

The parameters used in the simulation to model the contact between the screw bit and screw are given in Appendix A. These parameters were chosen using the guidelines outlined in Drake's documentation; extra care was taken to ensure parameters were not too large or too small, which would introduce numerical errors. 32 simulations were run in parallel to maximize compute resources and reduce the time taken to collect all of the data. A sample of the collected data is presented in Figure 3, which shows how the contact wrench varies when the offset along the x direction is varied.

Details of the different datasets collected, specifying the number of different x,y, and angular positions of the screw, and the number of x,y, and angular misalignments, is given in Table III-A. Additionally, the time taken to collect the datasets is given. The large datasets were used to train the contact wrench to misalignment models. After training, the models were tested on the small models.

B. Training the Misalignment Prediction Network

For each of the three tasks, the MLP was trained using the Adam optimizer with dynamic lowering of the learning rate when the validation loss plateaued [16]. The input contact wrench, joint angles, and end effector pose were all normalized before training. The network was trained for 200 epochs in the fixed screw case and 1000 epochs in the unfixed screw case.

Task	Train loss (m)	Validation loss (m)	Unseen data loss (m)
Fixed screw pose	$2.64 \cdot 10^{-6}$	$4.84 \cdot 10^{-6}$	$2.00 \cdot 10^{-8}$
Unfixed (X-Y) screw pose	0.0051	0.0048	$1.85 \cdot 10^{-5}$
Unfixed (X-Y-θ)	0.0034	0.0035	0.0071

TABLE II
TRAINING, VALIDATION, AND UNSEEN DATA LOSSES FOR THE MODELS TRAINED.

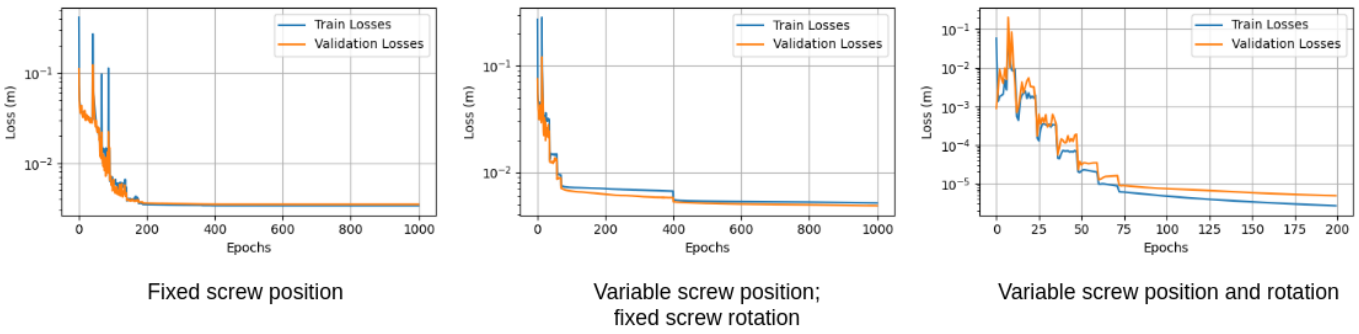

Fig. 4. Training and validation losses for the three different screw pose cases.

The loss curves for the networks are given in Figure 4. Values for the final training, validation, and unseen data losses are given in Table III-B. The models saved were the models from the final epoch of training.

C. Screw Fastening Task

A controller was developed that moved the robot end effector down onto the screw, measured the contact wrench, used the learned misalignment model to predict the screw's misalignment, and then raised and re-aligned the end effector and fastened the end effector's hex bit with the screw. To produce a noisy estimate of the screw pose given to the controller, noise was sampled and added to the known screw pose to produce an erroneous screw pose estimate. This erroneous screw pose estimate was given to the fastening controller, and the fastening controller had to correct this estimate using the learned network. This behavior replicates the process of correcting misalignment during a screw fastening task.

Screw Pose	Fastening Success Rate	Average number of fastening attempts
Fixed	97.5%	1.02
Unfixed (X-Y)	99%	1.95
Unfixed (X-Y-θ)	0%	N/A

TABLE III
RESULTS OF THE SCREW FASTENING TASK.

For each of the three cases, 200 trials of the fastening and re-aligning operation were run. A successful outcome is when the arm's end effector screw bit is aligned with the screw in less than 10 fastening attempts. Results specifying the percentage of successful trials and the average number of fastening attempts are specified in Table III-C.

IV. Discussion

In the simulation, the collected data for the contact matches what we would expect when fastening a hexagonal screw. In Figure 3, the z-component of the force is significantly larger than the other components. Since we are pushing downwards while fastening, we would expect the z-component of the force to be higher than the other components. Additionally, at larger x offsets, we see an increased reaction force in the x direction, which makes sense as the end effector would be tilted slightly toward the x axis thus causing a more pronounced x-component of the reaction force. Also, the sign of the x-component of the force matches what we expect: in negative offsets in the x-direction, the x-component of the force would be negative, and vice versa. These results in Figure 3 also match what previous works have found when measuring the contact wrench during fastening [8, Figure 5].

From the loss curves and low losses on unseen data, we can conclude that our networks are able to learn the contact wrench to misalignment relationship. Our loss curves all show the losses decreasing and plateauing, showing that our learning has found a local minimum for the network. Additionally, the unseen data losses are all low for the fixed screw pose and unfixed screw 2D position cases. However, for the case where the screw position and rotation is unfixed, the training and validation losses are substantially higher than for the other cases. This suggests that we have insufficient data points to learn the contact wrench to misalignment relationship - for the other cases we more misalignments in our training data, which enable those networks to learn the misalignment relationship better. One thing to note is that the training and validation losses for the unfixed screw position and fixed rotation case are higher than the unseen data loss. This may be due to anomalous data points which did not appear in the unseen data set. However, these high losses did not pose an issue for us, as this model was successful in executing the full fastening task.

The performance of our method on the actual fastening task is on par with previous works on misaligned peg-in-hole fastening [8, 9, 10]. Each of the cases where the fastening task failed were repeated to find the source of error. For the case with the fixed screw pose, it was found that the failures were caused by the cartesian controller exerting large forces during abrupt changes in the trajectory, namely when the end effector had to move down immediately after being raised up from the screw. These large forces led to numerical errors and crashes in Drake's simulation solver. In the case with the unfixed screw position, errors also occurred due to model inaccuracy, as the model could not predict with misalignment accurately enough to fasten successfully within 10 attempts. This also explains why more fastening attempts were needed to align

the screw compared to the fixed screw position case. The poor performance of our model on the case with an unfixed screw position and rotation is due to the model's inaccuracy: the loss of several millimeters during training exceeds the tolerance for the fastening task.

These results show that our method is capable of learning the contact wrench to misalignment relationship for the complex geometry of the hexagonal screw. This exceeds previous works which have not looked into learning this relationship for objects more complex than a cylindrical peg and hole [8, 9]. Additionally, collection of a large number of data points in 8 and 10 hours for the two successful tasks matches the time taken by prior works [8]. However, our ability to collect more data points than prior works is key in enabling us to learn the contact relationship for the complex hexagonal screw.

Our method's failure to extend to different screw rotations is due to having insufficient data points for the misalignment. From Figure 3, it is clear that the wrench to offset relationship is complicated and non-linear. Therefore, having only 11 different misalignments in the x, y, and rotational directions may be insufficient to learn the misalignment relationship. More misalignments could be tested in the training data simulation, but at the cost of a longer simulation time, which may be impractical for applications where this model must be re-trained quickly for other assembly tasks, or in situations where contact data can only be collected in the real world.

V. Conclusion

Our method is capable of successfully and reliably correcting misalignment during screw fastening both for a screw in a fixed position, and when the screw's 2D position is varied. Learning a contact wrench to misalignment relationship for the complex hexagonal screw exceeds previous work on correcting peg-in-hole misalignment by considering more complex object geometries [8]. Additionally, our work is successful in collecting many data points of high-quality fastening data in a similar time as previous works [8]. This shows the potential for supervised learning of misalignment and using synthetic training data for improving accuracy of common industrial assembly tasks.

However, our method falls short when adapting to different screw rotations due to the need to collect even more training data which requires longer time. This therefore suggests future research can investigate ways to learn the contact wrench to misalignment relationship with fewer data points, possibly by using a hybrid learning approach informed by an analytical model of the contact wrench. Another direction for future research is using these methods in real-world fastening tasks and exploring sim-to-real transfer of the contact data gathered in Drake.

To conclude, this work has developed a promising method for correcting misalignment in screw fastening tasks that shows potential for unstructured environments where the screw may be at different, unknown positions. The synthetic training data obtained in simulation is usable for learning a misalignment relationship, and our final models and controllers can achieve up to 99% success rates in our fastening tasks.

VI. References

[1] Yuze Jiang et al. "A review of robotic assembly strategies for the full operation procedure: planning, execution and evaluation". In: *Robotics and Computer-Integrated Manufacturing* 78 (2022), p. 102366. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2022.102366.

[2] Y. Litvak, A. Biess, and A. Bar-Hillel. "Learning Pose Estimation for High-Precision Robotic Assembly Using Simulated Depth Images". In: *2019 International Conference on Robotics and Automation (ICRA)*. Montreal, QC, Canada: IEEE Press, 2019, pp. 3521–3527. DOI: 10.1109/ICRA.2019.8794226. URL: https://doi.org/10.1109/ICRA.2019.8794226.

[3] Íñigo Elguea-Aguinaco et al. "A review on reinforcement learning for contact-rich robotic manipulation tasks". In: *Robotics and Computer-Integrated Manufacturing* 81 (2023), p. 102517. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2022.102517. URL: https://www.sciencedirect.com/science/article/pii/S0736584522001995.

[4] Ibrahim F. Jasim, Peter W. Plapper, and Holger Voos. "Position Identification in Force-Guided Robotic Peg-in-Hole Assembly Tasks". In: *Procedia CIRP* 23 (2014). 5th CATS 2014 - CIRP Conference on Assembly Technologies and Systems, pp. 217–222. ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2014.10.077.

[5] D. E. Whitney. "Quasi-Static Assembly of Compliantly Supported Rigid Parts". In: *Journal of Dynamic Systems, Measurement, and Control* 104.1 (Mar. 1982), pp. 65–77. ISSN: 0022-0434. DOI: 10.1115/1.3149634.

[6] Shuhan Li and Jinli Xu. "Multiaxis Force/Torque Sensor Technologies: Design Principles and Robotic Force Control Applications: A Review". In: *IEEE Sensors Journal* 25.3 (2025), pp. 4055–4069. DOI: 10.1109/JSEN.2024.3495507.

[7] Kuangen Zhang et al. "Jamming Analysis and Force Control for Flexible Dual Peg-in-Hole Assembly". In: *IEEE Transactions on Industrial Electronics* 66.3 (2019), pp. 1930–1939. DOI: 10.1109/TIE.2018.2838069.

[8] Devesh K. Jha et al. "Imitation and Supervised Learning of Compliance for Robotic Assembly". In: *2022 European Control Conference (ECC)*. 2022, pp. 1882–1889. DOI: 10.23919/ECC55457.2022.9838102.

[9] Tadanobu Inoue et al. "Deep reinforcement learning for high precision assembly tasks". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 819–825. DOI: 10.1109/IROS.2017.8202244.

[10] Hee-Chan Song, Young-Loul Kim, and Jae-Bok Song. "Automated guidance of peg-in-hole assembly tasks for complex-shaped parts". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, pp. 4517–4522. DOI: 10.1109/IROS.2014.6943202.

[11] Luigi Villani and Joris De Schutter. "Force Control". In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 161–185. ISBN: 978-3-540-30301-5. DOI: 10.1007/978-3-540-30301-5_8.

[12] Devesh K. Jha et al. "Design of Adaptive Compliance Controllers for Safe Robotic Assembly". In: *2023 European Control Conference (ECC)*. 2023, pp. 1–8. DOI: 10.23919/ECC57647.2023.10178229.

[13] O. Khatib. "A unified approach for motion and force control of robot manipulators: The operational space formulation". In: *IEEE Journal on Robotics and Automation* 3.1 (1987), pp. 43–53. DOI: 10.1109/JRA.1987.1087068.

[14] Russ Tedrake and the Drake Development Team. *Drake: Model-based design and verification for robotics*. 2019. URL: https://drake.mit.edu.

[15] Ryan Elandt et al. "A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 8238–8245. DOI: 10.1109/IROS40897.2019.8968548.

[16] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[17] Richard T. Barrett. *Fastener Design manual*. University Press of the Pacific, 2005.

VII. Appendix

A. Parameters used in Drake Screw Fastening Simulation

TABLE IV
PARAMETERS USED IN THE SIMULATION OF THE SCREW FASTENING OPERATION

Simulation time step	0.001 s
Penetration allowance	10^{-4} m
Contact Model	Hydroelastic
Stiction tolerance	0.01 m/s
Discrete Contact Solver	TAMSI
Proximity Margin	10^{-9} m
Resolution Hint	10^{-9} m
Hydroelastic modulus of screw and bit	$5 \cdot 10^{9}$ Pa
Hunt-Crossley Dissipation	10 s/m
Coefficient of Static Friction (steel-on-steel) [17]	0.7
Coefficient of Dynamic Friction (steel-on-steel) [17]	0.7

In the contact fastening simulation, the parameters used are given in Table IV. These parameters were chosen based off guidelines outlined in the Drake documentation. The parameters chosen gave suitable contact results while ensuring solve time remained acceptable and numerical errors due to extremely large or small parameters were avoided. The parameters are provided here simply as a means to replicate this work; for a more in-depth description of each parameter, consult the Drake documentation [14].